

Sommaire :

Sommaire :	1
Introduction.....	1
1 – Installation du serveur ProFTPD :	2
4 – Configuration générale :	4
5 – Analyse de trames FTP :	4
6 – ProFTP avec MySQL :	4
7 – Configuration avec KeepAlived :	6

Introduction

Objectif : Le FTP permet de transférer des fichiers d'un serveur vers des clients, et inversement. Le DTP est un processus qui gère le canal de données, et le PI est l'interpréteur de protocole.

On récupère les commandes en caractères. Le serveur les interprètes. En mode passif, le serveur donne un numéro de port au client, et ce dernier écoute sur ce numéro de port. C'est le client qui se connecte sur le serveur ensuite. Le mode passif est utilisé en cas de problème avec un pare-feu qui bloque les connexions.

Pré-requis : En mode actif, c'est le client qui initie la connexion sur le serveur FTP (port 21) et le client se met à écouter sur le port n+1, puis le serveur se connecte au client.

Grâce à Telnet, on peut utiliser des commandes à distance, sur le port 23 (TCP non chiffré).

Ports connus : inférieur à 1024

Ports réservés : supérieur à 1024

Ports libres : supérieur à 5000

Norme : Toutes les commandes issues d'une machine avec un système d'exploitation Debian ou Windows sont écrites *en gras et en italique*.

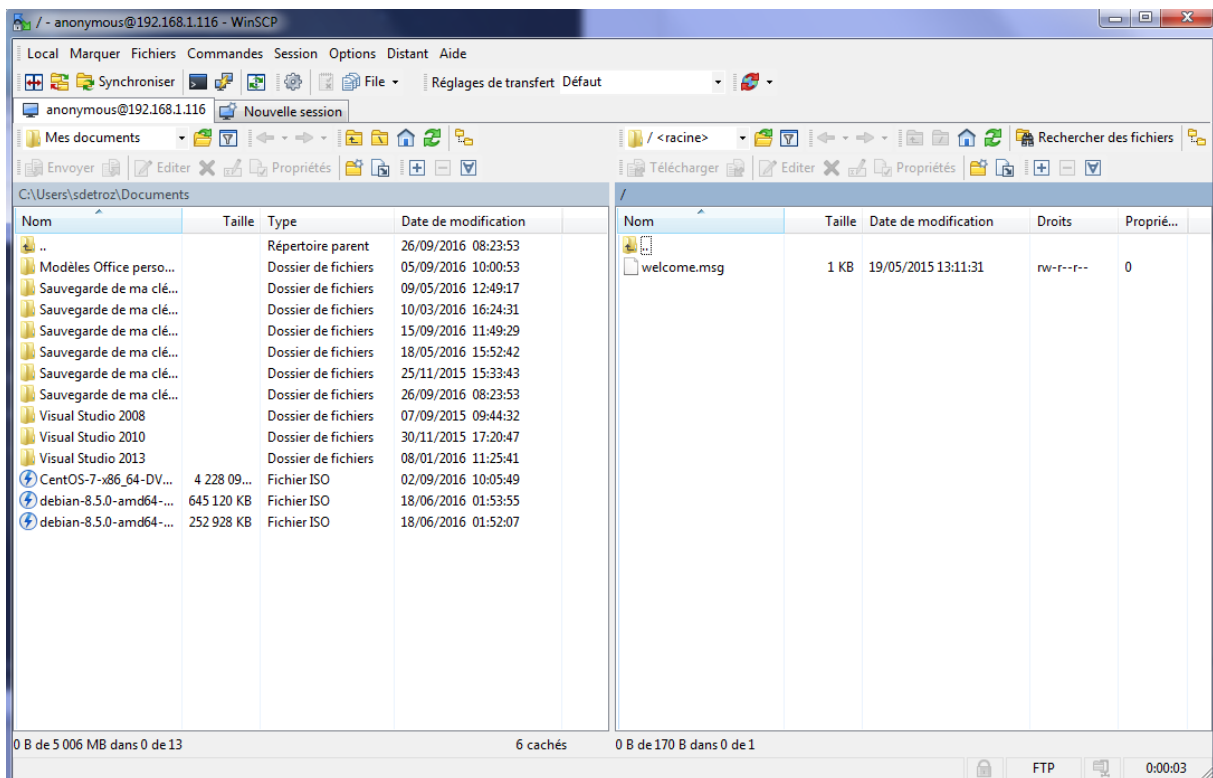
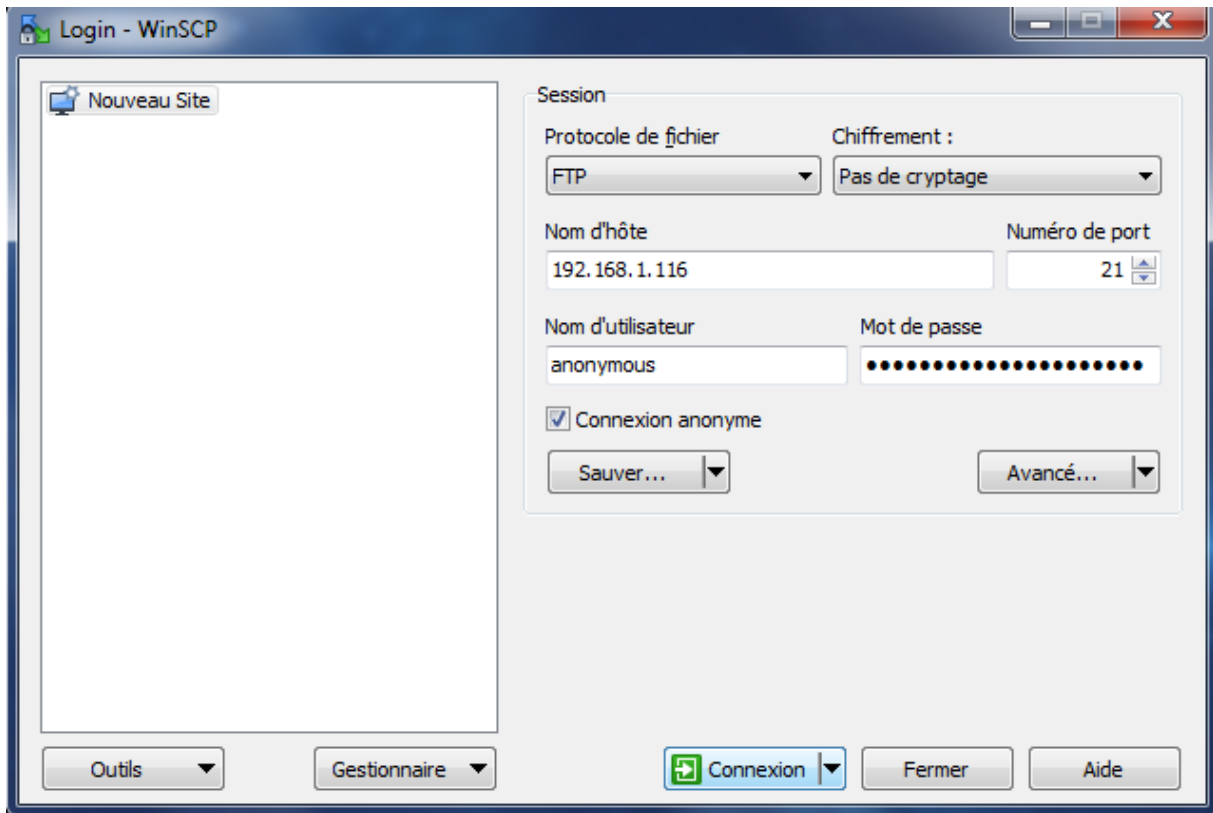
1 – Installation du serveur ProFTPD :

```
root@sebftp:~# apt-get install proftpd
```

Outil de configuration des paquets



```
root@sebftp:~# nano /etc/proftpd/proftpd.conf
```



4 – Configuration générale :

On installe Apache 2.

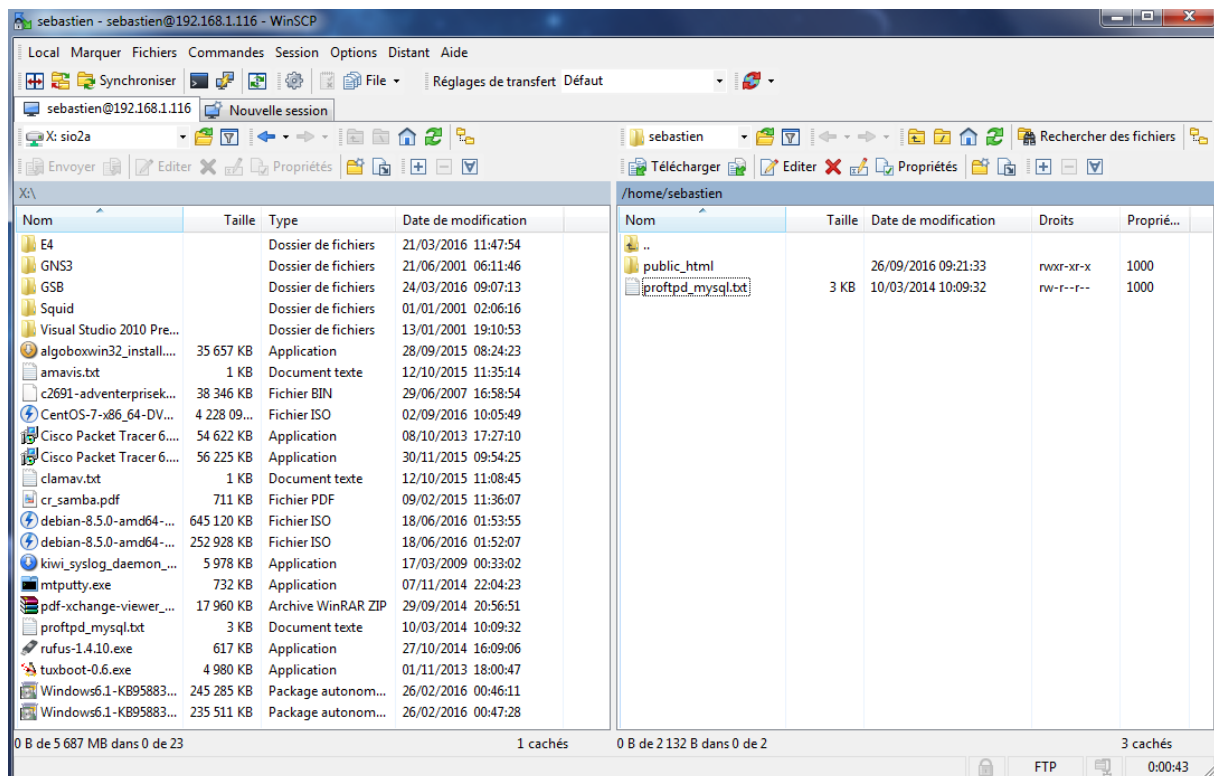
5 – Analyse de trames FTP :

La connexion FTP est pour l'échange d'informations entre le FTP et le client, et FTP-Data est pour l'échange de données. Ils utilisent des numéros de ports différents.

En mode passif, on additionne les deux derniers chiffres (dont le premier multiplié par 256) pour obtenir le numéro de port. Le mode passif, c'est le Serveur qui va donner un port au client pour pouvoir passer à travers le pare-feu depuis l'intérieur vers l'extérieur. Chaque client quant à lui fait une demande sur les ports 20 (données) et 21 (contrôles).

6 – ProFTP avec MySQL :

Installer ***apt-get install mysql-server*** et ***apt-get install proftpd-mod-mysql***.



mysql -u root -p (u pour user, p pour password)

On a ensuite accès à MySQL.

```
show databases ;  
create database proftpd ;  
show databases ;
```

```
grant select, insert, update, delete on proftpd.* to 'proftpd'@'localhost' identified by 'proftpd' ;
```

```
flush privileges ;
```

```
quit ;
```

On télécharge ensuite le fichier de base de données :

```
mysql -u root -p proftpd < /home/sebastien/proftpd_mysql.txt
```

On se connecte pour vérifier la création de la base :

```
mysql -u proftpd -p  
show databases ;  
use proftpd ;  
show tables ;  
desc ftpgroup ;
```

```
select * from ftpuser ;
```

Dans `/etc/proftpd/modules.conf`, on dé-commente :

```
LoadModule mod_sql.c  
LoadModule mod_sql_mysql.c
```

Dans `/etc/proftpd/sql.conf`, on dé-commente :

```
SQLBackend          mysql  
SQLEngin            on  
SQLAuthenticate      on  
SQLAuthTypes Crypt Plaintext (et on enlève Plaintext)
```

```
SQLConnectInfo proftpd@localhost proftpd proftpd
```

```
SQLUserInfo ftpuser userid passwd uid gid homedir shell  
SQLUserWhereClause « loginallowed = 'true' »
```

SQLGroupInfo ftpgroup groupname gid members**CreateHome on**

On créer ensuite le groupe FTP :

```
groupadd -g 5500 ftpgroup
useradd -u 5500 -s /bin/false -d /bin/null -g ftpgroup ftpuser
passwd -a ftpuser ftpgroup
```

On se connecte à la base :

```
mysql -u proftpd -p proftpd
insert into ftpgroup values ('ftpgroup',5500,'ftpuser');
select * from ftpgroup;
```

```
insert into ftpuser values (1,'seb',encrypt('seb'),5500,5500,'/home/seb','/sbin/nologin','','','');
select * from ftpuser;
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| id | userid | passwd          | uid | gid | homedir  | shell          | count |
|-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | seb    | qna1FCWoJHUn6 | 5500 | 5500 | /home/seb | /sbin/nologin | 0     |
| 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Il faut ensuite rajouter dans `/etc/proftpd/proftpd.conf` :

```
RequireValidShell    off
#SQLUserWhereClause « loginallowed = 'true' »
Include /etc/proftpd/sql.conf
```

On télécharge ensuite php5 et phpmyadmin.

[7 – Configuration avec KeepAlived :](#)

On va commencer par installer keepalived sur les deux machines :

```
apt-get install keepalived
```

On va ensuite configurer le fichier qu'on va créer et mettre le minimum d'information dedans :

```
nano /etc/keepalived/keepalived.conf
```

On va y ajouter cela :

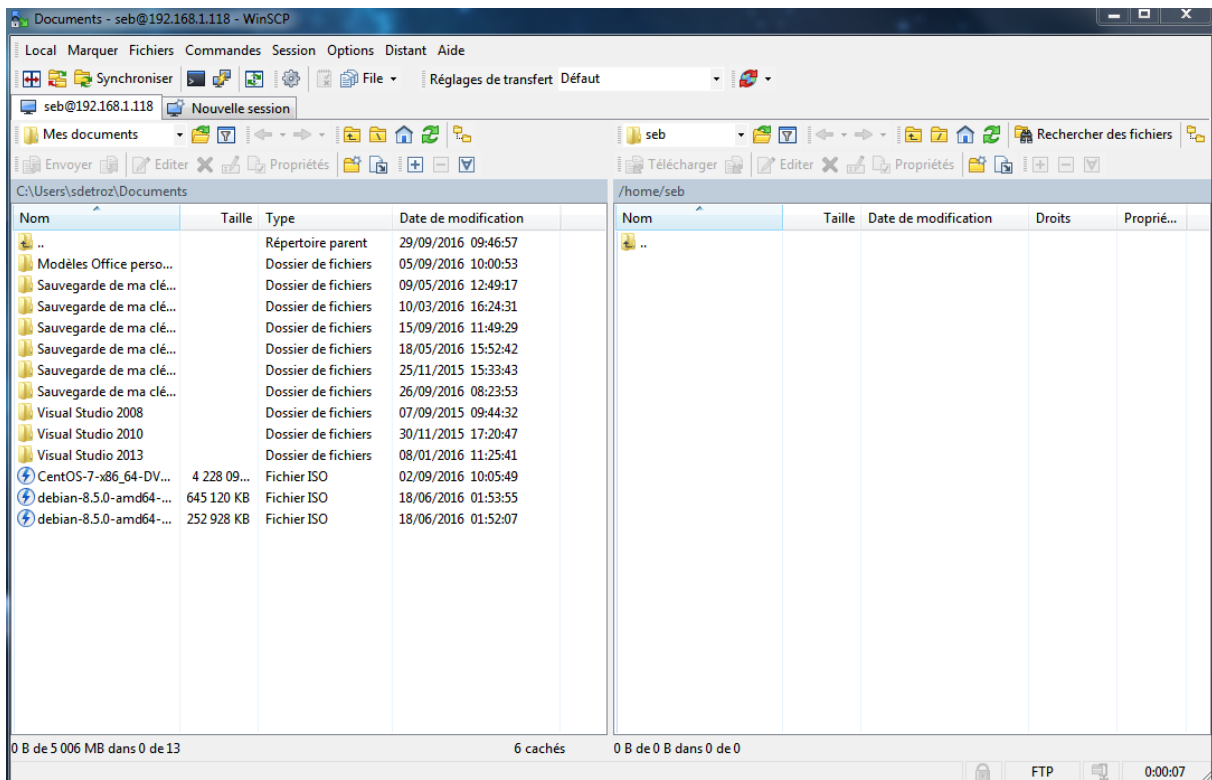
```
wrrp_instance FTPSEB {
    state BACKUP
    interface eth0
    virtual_router_id 44
    priority 100
    advert_int 5
    virtual_ipaddress {
        192.168.1.118
    }
}
```

Sur le deuxieme server on va mettre une priority 100

Ne pas oublier de redémarrer le service keepalived.

```
#systemctl restart keepalived
```

On va ensuite voir si on peut se connecter avec WINSCP avec les identifiants seb et seb avec l'adresse Ip virtuelle.



Cela marche correctement.

Ne pas oublier d'avoir des hostnames différents sur ces deux machines et d'avoir la correspondance dans le host :

```
#/etc/hostname    #/etc/hosts
```

Ne pas oublier de faire un restart !

Pour éviter que nos instances FTP entrent en conflit avec les autres VM de la classe, on utilise la connexion « Réseau interne » sur nos serveurs et sur une VM cliente pour pouvoir accéder à notre propre serveur FTP, ou on personnalise le nom d'instance et l'ID du router virtuel.

Lorsque tout est en place, on peut éteindre l'un ou l'autre des serveurs pour vérifier que la bascule s'effectue correctement. On peut par exemple voir que le premier serveur ProFTP devient serveur « maître » par défaut :

```
root@sebftp1:~# service keepalived stop
root@sebftp1:~# service keepalived start
root@sebftp1:~# tail /var/log/syslog
Sep 30 15:07:35 sebftp1 Keepalived_healthcheckers[2483]: Initializing ipvs 2.6
Sep 30 15:07:35 sebftp1 Keepalived_healthcheckers[2483]: Registering Kernel netlink reflector
Sep 30 15:07:36 sebftp1 Keepalived_healthcheckers[2483]: Registering Kernel netlink command channel
Sep 30 15:07:36 sebftp1 Keepalived_healthcheckers[2483]: Opening file '/etc/keepalived/keepalived.conf'.
Sep 30 15:07:36 sebftp1 Keepalived_healthcheckers[2483]: Configuration is using : 5009 Bytes
Sep 30 15:07:36 sebftp1 Keepalived_healthcheckers[2483]: Using LinkWatch kernel netlink reflector...
Sep 30 15:07:42 sebftp1 Keepalived_vrrp[2484]: VRRP_Instance(FTPSEB) forcing a new MASTER election
Sep 30 15:07:42 sebftp1 Keepalived_vrrp[2484]: VRRP_Instance(FTPSEB) forcing a new MASTER election
Sep 30 15:07:47 sebftp1 Keepalived_vrrp[2484]: VRRP_Instance(FTPSEB) Transition to MASTER STATE
Sep 30 15:07:52 sebftp1 Keepalived_vrrp[2484]: VRRP_Instance(FTPSEB) Entering MASTER STATE
```

```
root@sebftp2:~# service keepalived stop
root@sebftp2:~# service keepalived start
root@sebftp2:~# tail /var/log/syslog
Sep 30 15:07:25 sebftp2 Keepalived_vrrp[2452]: VRRP_Instance(FTPSEB) Entering BACKUP STATE
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Initializing ipvs 2.6
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Registering Kernel netlink reflector
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Registering Kernel netlink command channel
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Opening file '/etc/keepalived/keepalived.conf'.
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Configuration is using : 5007 Bytes
Sep 30 15:07:25 sebftp2 Keepalived_healthcheckers[2451]: Using LinkWatch kernel netlink reflector...
Sep 30 15:07:41 sebftp2 Keepalived_vrrp[2452]: VRRP_Instance(FTPSEB) Transition to MASTER STATE
Sep 30 15:07:41 sebftp2 Keepalived_vrrp[2452]: VRRP_Instance(FTPSEB) Received higher prio advert
Sep 30 15:07:41 sebftp2 Keepalived_vrrp[2452]: VRRP_Instance(FTPSEB) Entering BACKUP STATE
```


Et inversement lorsque le premier serveur est arrêté.

Pour la suite, il faut déclarer **les deux serveurs** dans le fichier `/etc/keepalived/keepalived.conf` :

```
virtual_server 192.168.1.118 80 {
    delay_loop 4
    lb_algo rr
    lb_kind DR
    persistence_timeout 120
    protocol TCP

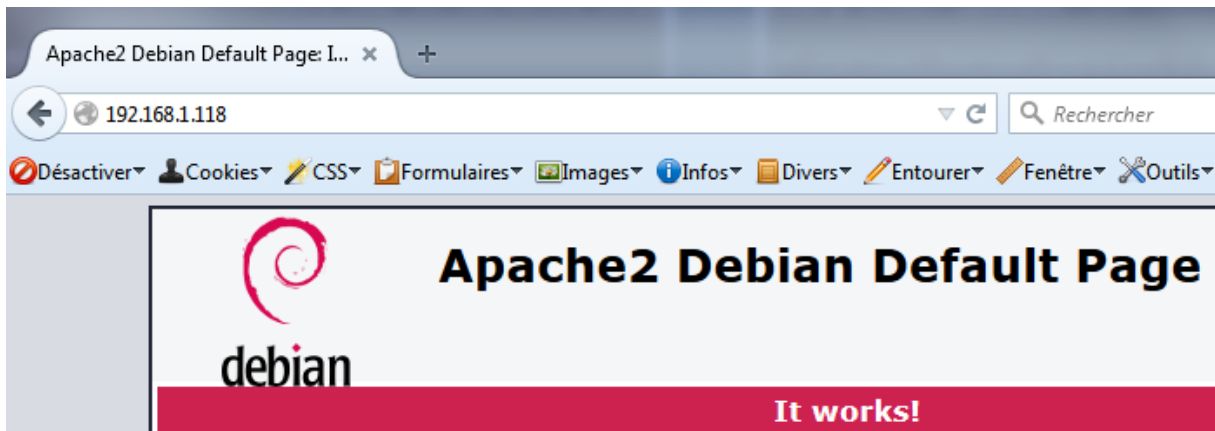
    real_server 192.168.1.116 80 {
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 5
        }
    }

    real_server 192.168.1.117 80 {
        weight 1
        TCP_CHECK {
            connect_port 80
            connect_timeout 5
        }
    }
}
```

Avec la commande `watch -n 3 -d ipvsadm -Ln`, on peut surveiller l'état de nos serveurs :

```
Every 3,0s: ipvsadm -Ln                               Fri Sep 30 16:21:58 2016
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  192.168.1.118:80 rr persistent 120
  -> 192.168.1.116:80        Route   1         0         0
  -> 192.168.1.117:80        Route   1         0         0
```

Lorsque tout fonctionne...



... on peut éteindre l'un de nos serveurs et essayer de se connecter à nouveau !